UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/699,062	10/31/2003	Bryan M. Cantrill	03226/330001; SUN040156	2597

32615 7590 09/10/2007

OSHA LIANG L.L.P./SUN
1221 MCKINNEY, SUITE 2800
HOUSTON, TX 77010

EXAMINER	
MYINT, DENNIS Y	

ART UNIT	PAPER NUMBER
2162	

MAIL DATE	DELIVERY MODE
09/10/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

mn

Office Action Summary	Application No.	Applicant(s)	
	10/699,062	CANTRILL, BRYAN M.	
	Examiner	Art Unit	
	Dennis Myint	2162	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 07/06/2007.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-10,12-17 and 19-24 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-10, 12-17, and 19-24 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This communication is responsive to Applicant's Amendment, filed on July 6, 2007.

2. The amendment filed on July 6, 2007 has been received and entered. Claims 1-10, 12-17, and 19-24 are pending in this application. Claims 11 and 18 have been cancelled. Claims 1, 6, 12, 15, 19, 23, and 24 are independent claims.

Response to Arguments

3. In light of the arguments filed on July 6, 2007, the Declaration filed on February 2, 2007, under 37 CFR 1.131 is considered to be sufficient to overcome the Tang reference (United States Patent Application Publication No. 2003/0217130). As such, new grounds of rejection are presented below.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-4, 6, and 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Pang et al., (hereinafter "Pang") (U.S. Patent Application Number 6493837) in view of Sharma et al., (hereinafter "Sharma") (U.S. Patent Number 5511190).

Referring to claim 1, Pang is directed to a method for obtaining data from a kernel (Pang, Figure 3, i.e., ***Data received from calling thread****?;* Pang Column 1 Lines 16-21, i.e., *One well known method of* ***tracking system performance data*** *is to use counters, which are stored* ***values*** *that get incremented every time a particular event occurs. For example****, an operating system could provide one counter for keeping track of disk reads*** *on a particular drive and another counter for keeping track of disk writes;* Pang Column 1 Lines 31-37, i.e., ***Event tracing*** *is a technique often used in conjunction with counters to* ***track system performance data*** *by recording events of interest in a log buffer. For example, a read on disk number one can be recorded as "read #1." The total number of reads can then be tallied in a post-processing phase by analyzing the logged data. On multiprocessor systems, log buffers also tend to be global entities;* and Pang Column 2 Lines 29-33, i.e., *The invention is generally realized as an event tracing program.* ***The event tracing program generally receives performance data about an event occurring on the computer system*** *from a data producer program. The event tracing program responds by recording the event performance data in one of a set of a log buffers).* Pang teaches obtaining data from kernel and storing said data in a buffer.

However, Pang does not explicitly teach the limitation: "(storing the data in a data set) in an aggregation buffer using an aggregation function".

On the other hand, Sharma teaches the limitation:

""(storing the data in a data set) in an aggregation buffer using an aggregation function" (Sharma, Column 2 Lines 44-47, i.e., *These hash-based techniques allow groupings and* ***aggregates to be generated*** *on the fly through the use of partial aggregates maintained in primary memory;* Sharma, Column 3 Lines 24-27, *the selected columns of database rows belonging to groups that can't fit into the group table* ***are buffered*** *then, when the buffer is full, written directly to an overflow disk file*).

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to modify the method of Pang to add the feature of storing data sets in an aggregation buffer using an aggregation function, as taught by Sharma, so that the resultant method would comprise storing the data in a data set) in an aggregation buffer using an aggregation function. One would have been motivated to do so because *there is a need for a grouping or aggregation process that, through efficient use of computer memory resources, can run largely in memory, yielding increased execution speeds* (Sharma, Column 2 Lines 22-25)

As per claim 2, Pang in view of Sharma teaches the limitation:

"wherein the data set comprises a key component" (Sharma, Column 2 Lines 63-67, i.e., *a matching procedure applies a hash function to the group identifier, generating* ***a hashed group value*** *that serves as an index into a memory-resident hash table that*

maps hashed group values into corresponding memory-resident group table entries; Sharma, Column 9 Lines 31-37, i.e., *After the first row is read, the grounding function applies the hash function HF 210 to the group identifier (i.e., "A10") associated with the first row. The resulting* ***hashed group value*** *(HF(A10)) serves as an index to an entry (HT[HF(A10)]) of the hash table HT 216, the contents of which contain the address (*GT[A10]) of the group table entry (if one exists) in which data for group A10 is being aggregated;* Also see Figure 2 of Sharma), "an aggregation identifier component" (Sharma Column 2 Lines 59-63, i.e., *For each row, values are picked up for select columns designated in a SQL group-by statement, including* ***a group value or identifier*** *from the group columns, and zero or more data values from the data columns;* Sharma, Column 2 Line 67 through Column 4 Lines 4, i.e., *Each group table entry stores for a single group (i.e.,* ***a unique group identifier****) aggregates built on the group members' selected data fields, a group identifier, and housekeeping data;* Sharma, Column 7 Lines 40-49, i.e., *The largest possible size of the Group Table is determined by* ***the number of unique values of the group columns*** *GC 252;* (Particular note that unique group identifier corresponds to "aggregation identifier" of the claim invention because said unique group identifier identifies each grouping caused by "group by" (i.e., aggregating function)), and "a value component" (Sharma Column 2 Lines 59-63, i.e., *For each row, values are picked up for select columns designated in a SQL group-by statement, including a group value or identifier from the group columns, and* ***zero or more data values from the data columns;*** Sharma, Column 6 Lines 37-31, i.e., *At least one of the selected columns SC 250 is designated as a group column*

GC 252. ***The remaining (zero or more) selected columns SC 250 are data columns DC 254, which provide the member data to be grouped or aggregated***).

As per claim 3, Pang in view of Sharma teaches the limitations:

"obtaining an expression" (Sharma, Column 9 Lines 31-37, i.e., *After the first row is read, the grounding function applies* ***the hash function HF 210*** *to the group identifier (i.e., "A10") associated with the first row. The resulting hashed group value (HF(A10)) serves as an index to an entry (HT[HF(A10)]) of the hash table HT 216, the contents of which contain the address (*GT[A10]) of the group table entry (if one exists) in which data for group A10 is being aggregated;* Note that a hash function is an expression;), "a new value" (Sharma, Column 6 Lines 37-31, i.e., *At least one of the selected columns SC 250 is designated as a group column GC 252.* ***The remaining (zero or more) selected columns SC 250 are data columns DC 254, which provide the member data to be grouped or aggregated***), and an aggregation identifier" (Sharma Column 2 Lines 59-63, i.e., *For each row, values are picked up for select columns designated in a SQL group-by statement, including* ***a group value or identifier*** *from the group columns, and zero or more data values from the data columns;* Sharma, Column 2 Line 67 through Column 4 Lines 4, i.e., *Each group table entry stores for a single group (i.e.,* ***a unique group identifier****) aggregates built on the group members' selected data fields, a group identifier, and housekeeping data;* Sharma, Column 7 Lines 40-49, i.e., *The largest possible size of the Group Table is determined by* ***the number of unique values of the group columns*** *GC 252;* (Particular note that unique group identifier

corresponds to “aggregation identifier” of the claim invention because said unique group identifier identifies each grouping caused by “group by” (i.e., aggregating function)),) (Note Figure 3 of Sharma, which depicts a continuous process which terminates only when the process teaches the end of file (EOF(T1) in Figure 3). In said process, aggregation functions/procedures are executed continuously. As such, new data value are obtained, a new expression (i.e., hash expression) is obtained, and a group identifier (i.e., aggregation identifier) is obtained each time an aggregation function/procedure is executed), and

“generating a key using the expression and the aggregation identifier” (Sharma, Column 2 Lines 63-67, i.e., *a matching procedure* ***applies a hash function to the group identifier, generating a hashed group value*** *that serves as an index into a memory-resident hash table that maps hashed group values into corresponding memory-resident group table entries;* Sharma, Column 9 Lines 31-37, i.e., *After the first row is read, the grounding function* ***applies the hash function HF 210 to the group identifier*** *(i.e., "A10") associated with the first row. The resulting* ***hashed group value*** *(HF(A10)) serves as an index to an entry (HT[HF(A10)]) of the hash table HT 216, the contents of which contain the address (*GT[A10]) of the group table entry (if one exists) in which data for group A10 is being aggregated;* Also see Figure 2 of Sharma; Note that a key (i.e., a hashed group value) is generated using the expression (a hash function) and the aggregation identifier (i.e., group identifier).

As per claim 4, Pang in view of Sharma teaches the limitations:

"wherein the storing the data comprises:

"storing the key in the key component" (Sharma, Column 2 Lines 63-67, i.e., *a matching procedure applies a hash function to the group identifier, generating* ***a hashed group value*** *that serves as an index into a memory-resident hash table that maps hashed group values into corresponding memory-resident group table entries;* Note figure 2 of Sharma wherein the key (hashed group value) is stored in HT216);

"storing the aggregation identifier in the aggregation identifier component" (Sharma, Column 7 Lines 40-49, i.e., *The largest possible size of the Group Table is determined by* ***the number of unique values of the group columns*** *GC 252;* Also note Sharma's figure 2 where aggregation identifier (GC252) is stored in a column of the table;); and

"updating the current value in the value component using the new value and the aggregation function" (Sharma, Column 3 Lines 6-10, i.e., *When a corresponding group table entry exists, an aggregation procedure aggregates the new data values into the group data fields of that group table entry and updates housekeeping data of the group table entry*).

Referring to claim 6, Pang et al. in view of Sharma is directed to a method for storing data in a data set, wherein the data set comprises a key component (Sharma, Column 2 Lines 63-67, i.e., *a matching procedure applies a hash function to the group identifier, generating* ***a hashed group value*** *that serves as an index into a memory-resident hash table that maps hashed group values into corresponding*

memory-resident group table entries; Note figure 2 of Sharma wherein the key (hashed group value) is stored in HT216), an aggregation identifier component (Sharma, Column 7 Lines 40-49, i.e., *The largest possible size of the Group Table is determined by* ***the number of unique values of the group columns*** *GC 252*), and a value component (Sharma Column 2 Lines 59-63, i.e., *For each row, values are picked up for select columns designated in a SQL group-by statement, including a group value or identifier from the group columns, and* ***zero or more data values from the data columns***"), comprising:

"obtaining an expression, a new value, and an aggregation identifier" (Sharma, Column 9 Lines 31-37 as applied to claim 3 above; Sharma, Column 6 Lines 37-31 as applied to claim 3 above; and Sharma, Column 2 Line 67 through Column 4 Lines 4 as applied to claim 3 above);

"generating a key using the expression and the aggregation identifier" (Sharma, Column 2 Lines 63-67, i.e., *a matching procedure* ***applies a hash function to the group identifier***, ***generating a hashed group value*** *that serves as an index into a memory-resident hash table that maps hashed group values into corresponding memory-resident group table entries;* Sharma, Column 9 Lines 31-37, i.e., *After the first row is read, the grounding function* ***applies the hash function HF 210 to the group identifier*** *(i.e., "A10") associated with the first row. The resulting* ***hashed group value*** *(HF(A10)) serves as an index to an entry (HT[HF(A10)]) of the hash table HT 216, the contents of which contain the address (*GT[A10]) of the group table entry (if one exists) in which data for group A10 is being aggregated;* Also see Figure 2 of Sharma; Note

that a key (i.e., a hashed group value) is generated using the expression (a hash function) and the aggregation identifier (i.e., group identifier)); and

"storing the data set in a buffer" (Sharma, Column 2 Lines 44-47, i.e., *These hash-based techniques allow groupings and* ***aggregates to be generated*** *on the fly through the use of partial aggregates maintained in primary memory;* Sharma, Column 3 Lines 24-27, *the selected columns of database rows belonging to groups that can't fit into the group table* ***are buffered*** *then, when the buffer is full, written directly to an overflow disk file*), "wherein storing the data set comprises storing the key in the key component, storing the aggregation identifier in the aggregation identifier component" (Pang in view of Sharma, as applied to claim 3 above) and "updating a current value in the value component using the new value and an aggregation function" (Sharma, Column 3 Lines 6-10, i.e., *When a corresponding group table entry exists, an aggregation procedure aggregates the new data values into the group data fields of that group table entry and updates housekeeping data of the group table entry*).

Referring to claim 10, Pang in view of Sharma is directed to the method of claim 6, and teaches the limitation:

"wherein the expression comprises an n-tuple" (Sharma, Column 2 Lines 63-67, i.e., *a matching procedure* ***applies a hash function to the group identifier, generating a hashed group value*** *that serves as an index into a memory-resident hash table that maps hashed group values into corresponding memory-resident group table entries;* Sharma, Column 9 Lines 31-37"). In the method of claim 2 as taught by

Pang in view of Sharma, aggregation functions would be applied to the data from the kernel-level (as taught by Pang). n-tuples are inherent is said data from the kernel-level because said data could comprise data from different applications or client computers.

6. Claims 5, 7-9, 12, 13, 15, 16, 17, 20 and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Tang et al. in view of Coss et al. and further in view of Larson et al., (hereinafter "Larson") (U.S. Patent Number 6578131).

Referring to claim 5, Pang in view of Sharma as applied to claim 4 does not explicitly disclose the limitations of claim 5. However, Larson teaches a method and system for scalable hash table wherein data is stored employing a hash table, comprising:

generating a hash key (Larson et al., Column 6 Line 40-55, i.e. "a hash table". Note that any hash table generates a hash key);

searching for a hash bucket (Larson et al., Column 6 Line 40-55, i.e. "lookup");

searching for a hash chain element in the hash bucket (Larson et al., Column 7 Line 66 through Column 7 Line 28, i.e. "hash chain");

updating the value component of the data set associated with the hash chain element if a hash chain element corresponding to the key is found (Official note is taken that the concept of hash table and updating data using a hash-table search is notorious well know in the art.),

creating a new hash chain element if the hash chain element corresponding to the key is not found and updating the value associated with the new hash chain element

(Official note is taken that creating new hash chain is in a hash table is notorious well know in the art in any hashing algorithm).

At the time the invention was made, it would have been obvious to a person of ordinary kill in the art to modify the method of Pang in view of Sharma to add the feature of employing a hash table to update and lookup data as taught Larson to the method of Pang in view of Sharma so that, in the combined method, storing the data set further comprises:

generating a hash key using the key (Larson in view of Sharma);

searching for a hash bucket corresponding to the key using the hash key (Larson in view of Sharma);

searching for a hash chain element in the hash bucket corresponding to the key; (Larson in view of Sharma)

updating the value component of the data set associated with the hash chain element if a hash chain element corresponding to the key is found, wherein the updating the value component comprises applying the aggregation function to the current value in the value component using the new value as input (Larson in view of Sharma);

creating a new hash chain element if the hash chain element corresponding to the key is not found, wherein creating a new hash chain element comprises associating a new

> data set with the new hash chain element, storing the key in a key component of the new data set, storing the aggregation identifier in an aggregation identifier component of the new data set, and storing an initial

value in a value component of the new data set (Larson in view of Sharma); and

updating the value component associated with the new hash chain element, wherein the updating the value component associated with the new hash chain element comprises

applying the aggregation function to the initial value using the new value as input (Larson in view of Sharma).

One would have been motivated to add the feature of employing hash table for data update and lookup as taught by Larson et al. because "hashing is often used to provide fast lookup of items in a cache" (Larson et al., Column 1 Line 22-28).

Claim 7 is rejected on the same basis as claim 5.

Referring to claim 8, Pang in view of Sharma and further in view of Larson is directed to the method of claim 7, wherein the hash chain element is associated with the data set using a pointer (Larson et al. Column 7 Line 66 through Column 7 Line 13, i.e. "*Each entry in the hash table consists of nothing more than a pointer to a linked list containing all items hashing to that address. The linked list, or "hash chain", can either be embedded or separate. In the first case, the pointer field needed for the hash chain is embedded in the items themselves,..*"). Therefore, in the combined method of Pang in view of Sharma and further in view of Larson said pointer would be pointing to the data set.

Referring to claim 9, Pang in view of Sharma and further in view of Larson is directed to the method of claim 7, wherein the new hash chain element is associated with the new data set using a pointer. As applied in claim 8 above, in the combined method of Pang in view of Sharma and further in view of Larson, the new hash chain element is associated with the new data set using a pointer.

Claim 12 is rejected on the same basis as claim 7. Claim 7 incorporates all the limitations of claim 6.

As per claim 13, Pang in view of Sharma and further in view of Larson teaches the limitation:

"wherein the user-level table is a hash table". (Both Sharma and Larson teaches hash tables which are in the user-level).

Claim 15 is essentially the same as claim 9 except that it set forth the claimed invention as a data aggregation buffer rather than a method for storing data in a data set and rejected for the same reasons as applied hereinabove. Note that dependent claim 9 incorporates all the limitations of claim 6 and 7.

As per claim 16, Pang in view of Sharma and further in view of Larson teaches the limitation:

"wherein the key is generated from an function using an aggregation identifier" (Sharma, Column 2 Lines 63-67, i.e., *a matching procedure* ***applies a hash function to the group identifier, generating a hashed group value*** *that serves as an index into a memory-resident hash table that maps hashed group values into corresponding memory-resident group table entries*).

As per claim 17, Pang in view of Sharma and further in view of Larson teaches the limitation:

"where in the key is generated from a function using an aggregation identifier and an expression" (Sharma, Column 2 Lines 63-67, i.e., *a matching procedure* ***applies a hash function to the group identifier, generating a hashed group value*** *that serves as an index into a memory-resident hash table that maps hashed group values into corresponding memory-resident group table entries*).

Claim 24 is essentially the same as claim 9 except that it set forth the claimed invention as a computer system one network obtaining data from a kernel rather than a method for obtaining data from a kernel and rejected for the same reasons as applied hereinabove.

7. Claim 14, 19, and 20-23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Pang in view of Sharma and further in view of Larson and further in view of Barnett (U.S. Patent Application Publication Number 2003/0159132).

Referring to claim 14, Pang in view of Sharma and further in view Larson as applied to claim 5 above does not explicitly recite the use of dictionaries. However, using data dictionaries along with hash tables is well known in the art. For instance, Barnett teaches a method and system for conformance checking, wherein data dictionaries are used along with hash tables (Barnett et al. Paragraph 0028 and 0040).

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to modify the method of Pang in view of Sharma and further in view of Larson to add the feature of using data dictionaries along with hash tables as taught by Barnett to the method of Pang in view of Sharma and further in view of Larson so that in resultant method and system would be directed to the method claim 12, wherein obtaining the aggregation identifier matching the value of the aggregation identifier comprises search at least one selected from the group consisting of a user-level dictionary and a kernel level dictionary. Note that Tang et al. teaches data from user-level and kernel-level. One would have been motivated to do so in order to perform retrieval operations in a hash method, wherein key-value pairs are collected together as a dictionary (Barnett, Paragraph 0040).

Claim 19 is essentially the same as claim 14 except that it set forth the claimed invention as a data aggregation system rather than a method for storing data in a data

set and rejected for the same reasons as applied hereinabove. Note that Pang teaches both user-level and kernel-level data.

Claim 20 is essentially the same as claim 15 except that it set forth the claimed invention as a data aggregation system rather than an aggregation buffer and rejected for the same reasons as applied hereinabove. Note that Pang teaches both user-level and kernel-level data.

Claim 21 is essentially the same as claim 14 except that it set forth the claimed invention as a data aggregation system rather than a data aggregation buffer and rejected for the same reasons as applied hereinabove.

Claim 22 is essentially the same as claim 17 except that it set forth the claimed invention as a data aggregation system rather than a data aggregation buffer and rejected for the same reasons as applied hereinabove.

Claim 23 is essentially the same as claim 14 except that it set forth the claimed invention as an apparatus for obtaining data from a kernel rather than a method and rejected for the same reasons as applied hereinabove.

Contact Information

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dennis Myint whose telephone number is (571) 272-5629. The examiner can normally be reached on 8:30AM-5:30PM Monday-Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Breene can be reached on (571) 272-4107. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Dennis Myint
Examiner
AU-2162

JOHN BREENE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100